LA-UR-21-24248
Approved for public release; distribution is unlimited.

Title: The iRage Cookbook

Author(s): Mercer-Smith, James Alastair

Intended for: In-line help package

Issued: 2021-05-03

The iRage Cookbook

Jas. Mercer-Smith, XTD-PRI

Abstract

iRage is a family of Zsh and Python 3 scripts designed to accelerate the process of submitting calculations for the novice xRage user.

iRage reduces the time to write a new input deck and submit an xRage problem to the production queue to about 15 minutes, assuming the user has defined the initial geometry using a program like Osito or linked to problem geometries generated by codes such as Abaqus, Flag, or Pagosa.

iRage runs on Linux and Mac systems.

Los Alamos National Laboratory Is operated by Triad National Security, LLC for the U.S. Department of Energy National Nuclear Security Administration

The iRage command line; e.g., “iRage new=myjob now”

- If the iRage user sources the EAP .cshrc file, then the command “iRage” is already aliased; otherwise, the user can set an alias:

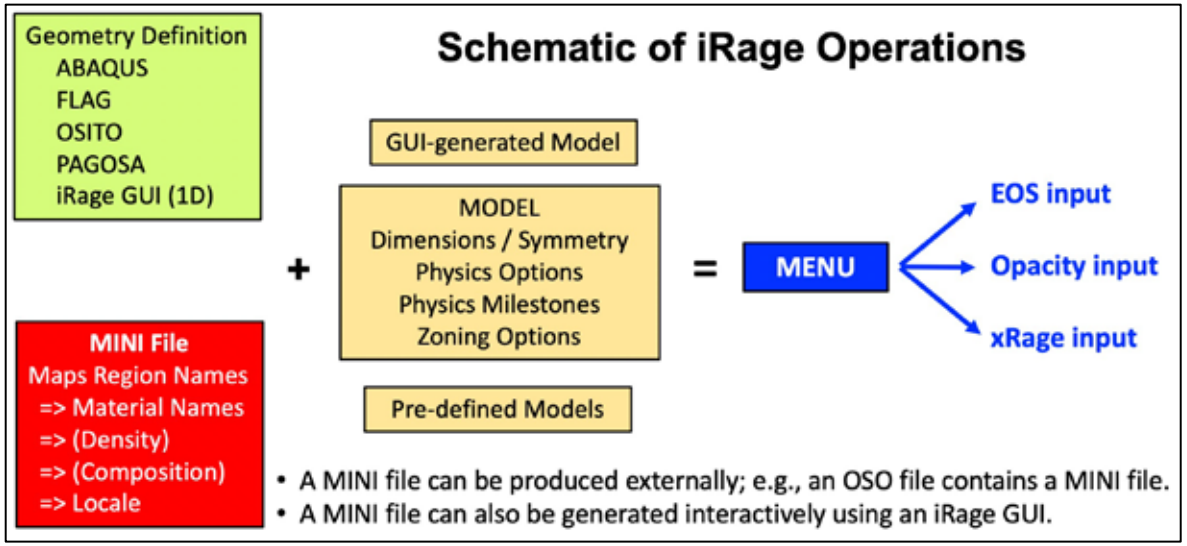
  iRage ‘zsh /usr/projects/eap/tools/iRAGE/iRage.zsh’

- While iRage is a Zsh script that calls Python 3, it was designed to be run from the default Cshell.

- The iRage command line has two arguments. The first argument triggers iRage options. The second argument defines how a job is submitted to the production queue. For example for problem arbitrarily named ”myjob”, the command line “iRage new=myjob now” will:
  - Guide the user through GUIs that offer a variety of physics options. Generate EOS and Opacity tables. Produce the rage.input deck. Create a log file. Store the files in the /scratch/moniker/myjob directory, and submit an xRage job to the production queue.
- The iRage toolkit has other useful capabilities, including: analytic contour generation, material properties lookup, and the calculation of normalized isotopic mixtures.
- Command line options are described in the help package called by the command “iRage help”.

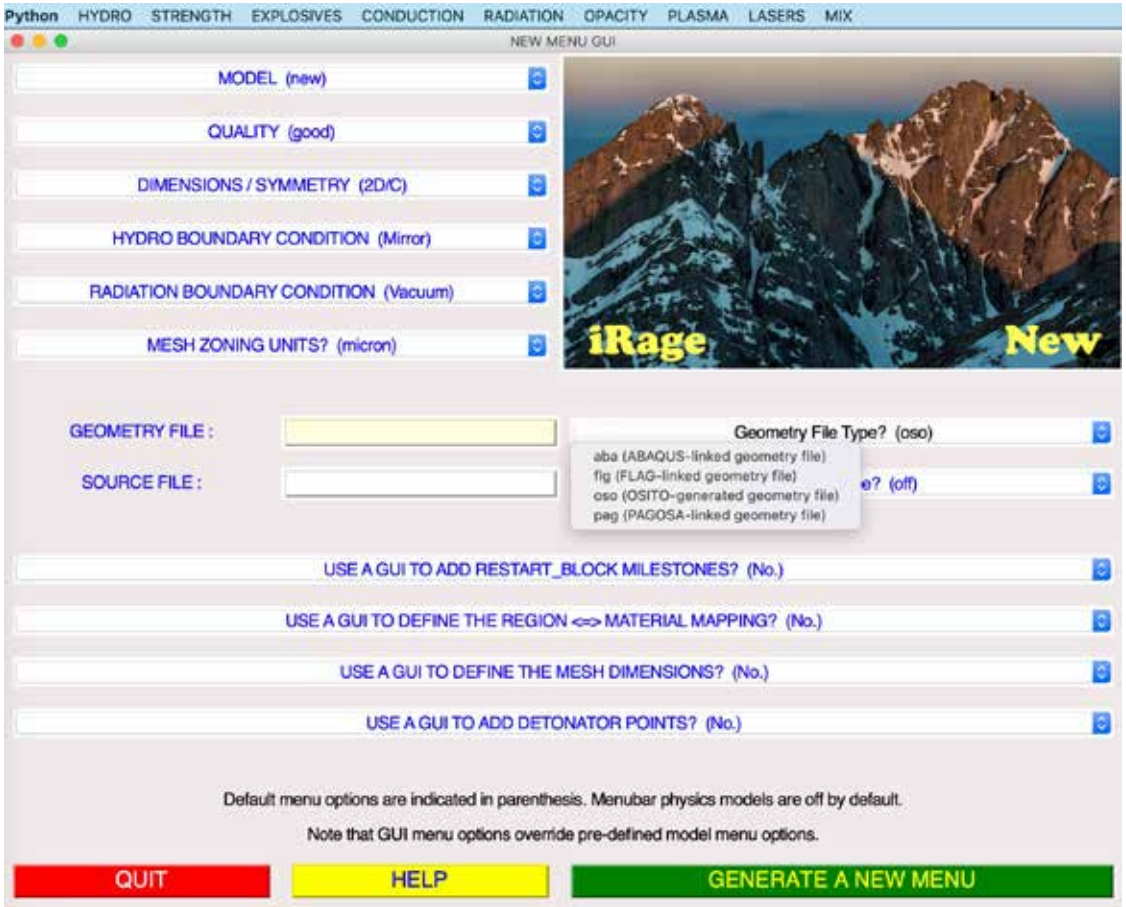
iRage assembles input decks from MODEL and MINI files.

- A MODEL is a set of physics model options and milestones that specify when physics approximations should be turned on or off.
- A MODEL can be pre-defined or iRage can generate a MODEL interactively using a GUI.



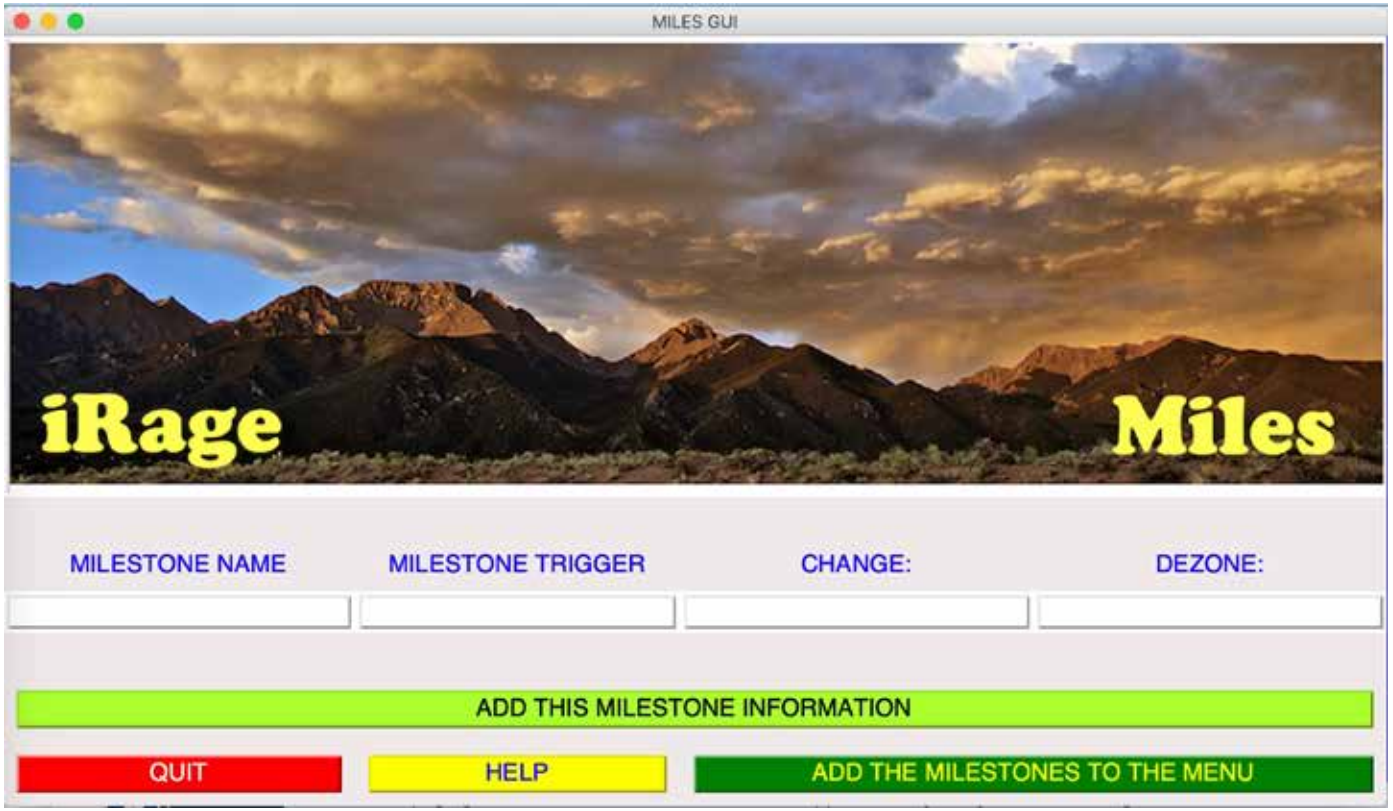
- A MINI file contains: (1) the path to the geometry file, (2) the dimensions of the hydrodynamic mesh, and (3) a mapping from “Region” names (aka, parts) to ”Material” names.
  - The Region <=> Material mapping, which includes the options for non-default densities and compositions, can be generated externally by a tool like Osito or internally with an iRage GUI.
- iRage combines a MODEL with a MINI file to produce a MENU file, which has all of the information necessary to generate an xRage input deck.

iRage has GUIs that can create new xRage input decks.



- When generating a new problem, iRage calls the “New” GUI which has dropdown menus to define physics model options, the problem geometry, and triggers for additional GUIs that define the mesh, the region-material mapping, and any restart milestones.
- iRage can extract information about the mesh dimensions and the the region <=> material mapping from a MINI file or an OSO file if either exists. This may circumvent the need to call the “Mesh” GUI and/or or the “Map” GUI.
- Specifying the geometry and a pre-defined model provides sufficient information to submit a new xRage simulation from the iRage command line; e.g., ‘iRage new.mod=myjob.oso new’.

The xRage restart block is a powerful tool. The iRage "Miles" GUI can set restart blocks at specified milestones.

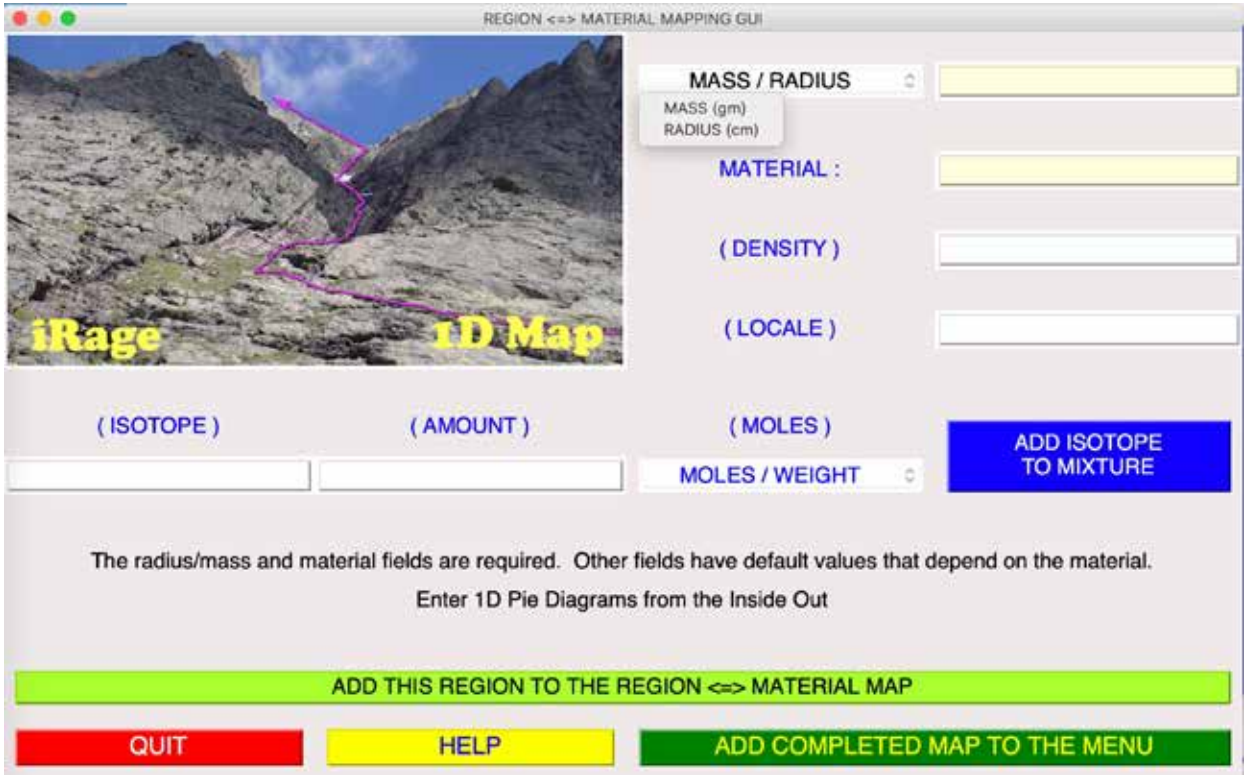


- The milestone name is arbitrary. The milestone trigger defines the xRage restart block; e.g., "time .gt. 47". The change command inserts an xRage command or triggers an iRage physics model option; e.g., "_radiation on". The dezone command specifies a locale and a power of two dezoning factor; e.g., "DET 4".

The iRage "Map" GUI defines the regions ⇔ materials map interactively. This mapping produces a MINI file.

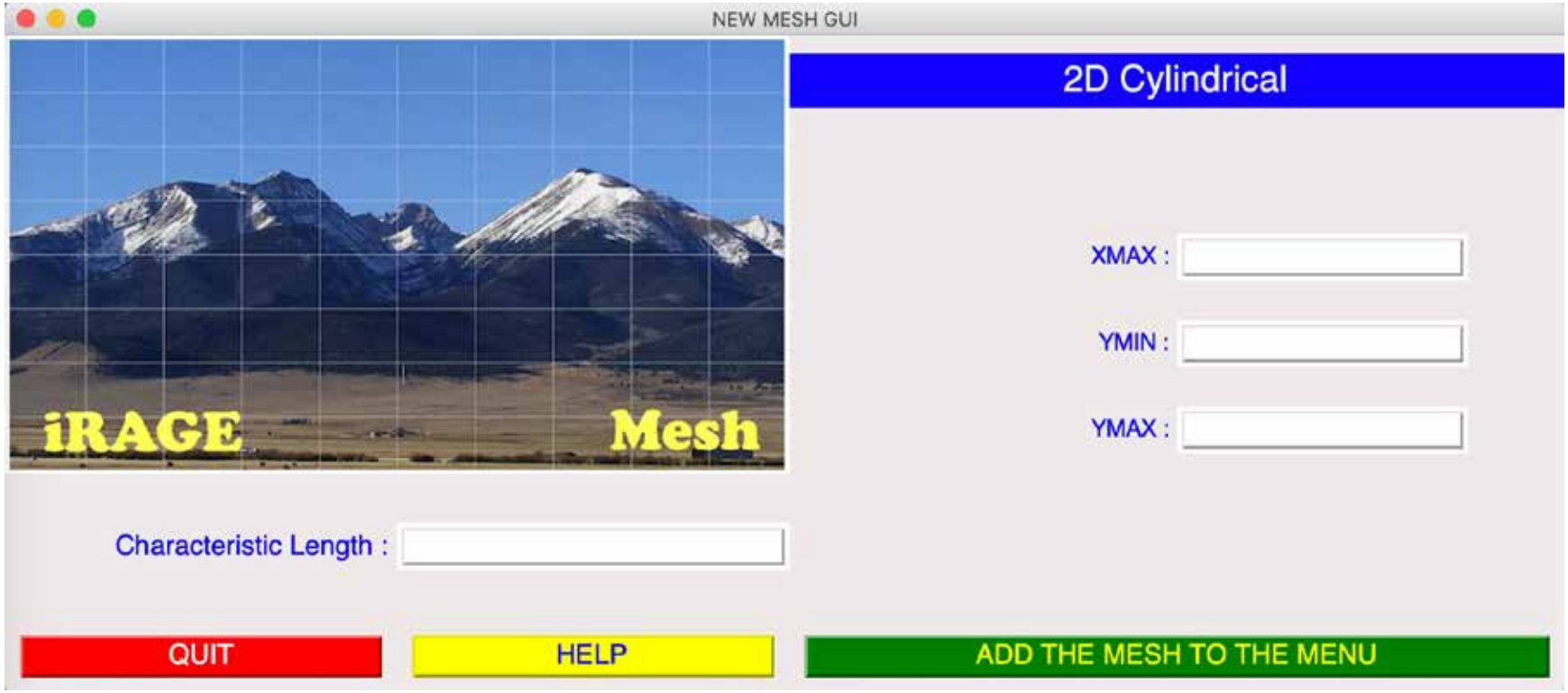
- Each Region ⇔ Material mapping permits the user to override the default density, and material composition.
- The completed map is a MINI file. iRage then generates a MENU file, as well as the "rage.input" deck whether or not the user has specified a geometry file.

The iRage “1D Map” GUI builds pie diagrams interactively.



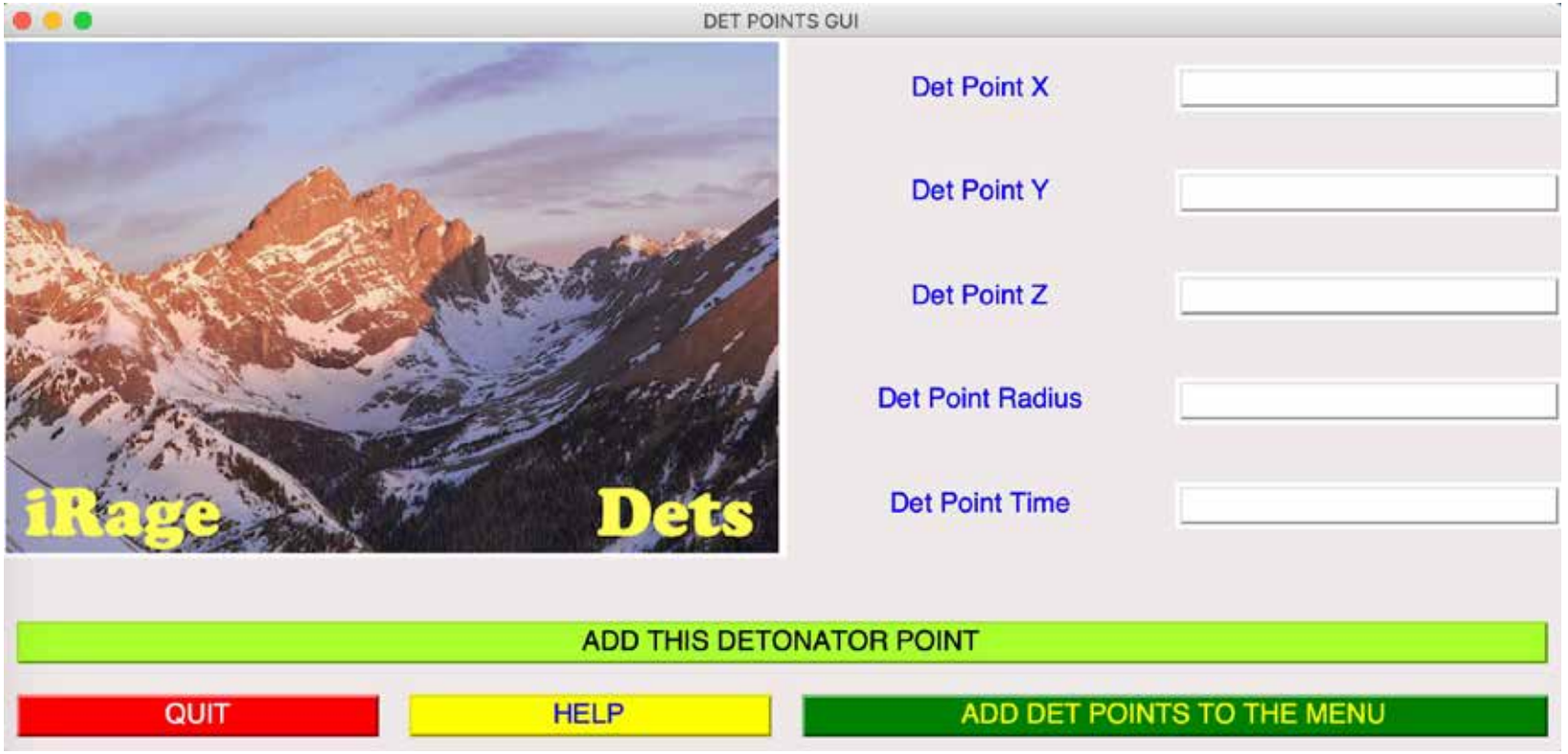
- In 1D, the region names are defined by the outer radius of each shell. The user has a dropdown menu to select either the mass or the outer radius for each shell.
- The pie diagram with both radii and masses is stored in the 1D MINI file.

The iRage “Mesh” GUI defines the hydro mesh dimensions.



- The “Dimensions / Symmetry” option on the “New” GUI page defaults to ”2D Cylindrical”.
- iRage calculates the optimal Level 1 mesh size automatically.

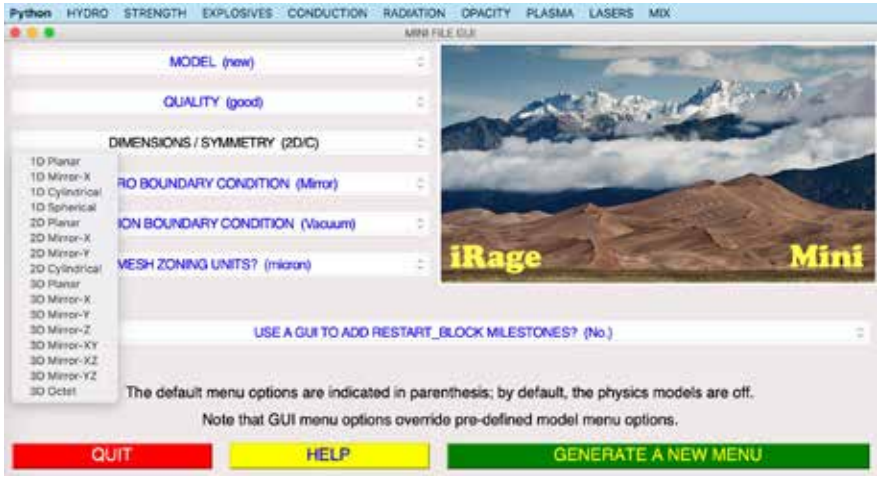
The iRage “Dets” GUI sets HE detonation point locations.



- The iRage “Dets” GUI page is only called if the user selected a high explosive model on the “New” GUI page.
- iRage has the option to add as many detonation points as the user may desire.

A MINI file identifies the geometry and region ⇔ material map; the “Mini” GUI can add the physics MODEL options.

```
# ******************                tmp37.mini                ******************
*// project:     tmp37
*// model:       jas                    ! Pre-defined model for testing
*// geofile:     test.oso               ! oso (OSITO-generated geometry file)
BEGIN REGINFO
reg1            copper          3 0.025 1.0e6 0 1 LOC // !=Default copper, @=Part 123,
202             salt            3 0.025 1.0e6 0 1 LOC // !=Not iodized, %m na 0.49 cl 0.49 %
reg_charge:B    pbx9501:B       3 0.025 1.0e6 0 1 HEX // !=Default PBX-9501 charge B, @=Part 789B,
reg52           polystyrene:52  3 0.025 1.0e6 0 1 LZ2 // rho=.52, %m h 0.52 c 0.48 %
BEGIN BBOX    0.0  127.0 -127.0  127.0    0.0    0.0    0.0 // xmin, xmax, ymin, ymax, zmin, zmax, clength
```



- The information in a MINI file, combined with physics options and a geometry file, is both necessary and sufficient to uniquely define the input deck for an xRage simulation.
- The format of a ‘.mini” file is the same as the REGINFO block of an OSO file.
- “iRage mini=myjob” can use a MINI file to link Abaqus, Flag, or Pagosa calculations to xRage.
- “iRage mini.new=myjob” calls a GUI to allow for the interactive selection of physics options, while “iRage mini.mod=myjob” invokes the pre-defined options of the “mod” model.

Pre-defined models provide the iRage user with shortcuts; e.g., “iRage new.mod=myjob.oso now”.

```
# ********************************************************************************
# ****************              iRmod_jas.txt              ******************
# ********************************************************************************
#
*// model:         jas                    ! A model for testing purposes
*// quality:       good                   ! Default physics options and zoning
*// geometry:      2D Cylindrical         ! Dimensions / Symmetry
*// hydrobc:       Mirror                 ! Hydro Boundary: Mirror, Vacuum
*// radbc:         Vacuum                 ! Rad Boundary: Mirror, Vacuum, T(time)
*// units:         micron                 ! Mesh zoning units
*//
*// hydro:         ip6                    ! Options: ip0, ip1, ip2, ip4, ip6, vof
*// strength:      sg                     ! Options: off, ep, jc, kos, ptw, ptw1, sg
*// explosives:    cg_mie.jwl             ! Options: off, (cg,ff,hi,so,sp).(hom,jwl,mie,
*// conduction:    spitzer                ! Options: off, sesame, spitzer
*// radiation:     gd                     ! Options: off, gd, mgd
*// opacity:       opc20_lte              ! Options: off, opc(20,25,30,58)_(lte,nlte)
*// plasma:        3t                     ! Options: off, 2t, 3t
*// lasers:        off                    ! Options: off, lle, nif
*// mix:           off                    ! Options: off, bhr2, bhr3, modal
*//
*// tstop:         1.0e-4                 ! Forced stop time, tmax = 1.0e-4.
&//
&// milestone:     time .ge. 0.0          ! Begin simulation @ start
&// change:        par = 0.0              ! Set parameter par @ start
&// change:        par2 = 0.0             ! Set parameter par2 @ start
&// change:        _radiation on          ! Turn on radiation @ start
&//
&// milestone:     time .gt. 1            ! Trigger restart milestone @ mile1
&// change:        par = 1.1              ! Change parameter par @ mile1
&// change:        _conduction            ! Turn on conduction @ mile1
&// change:        _radiation             ! Turn on specified radiation model @ mile1
&// change:        _plasma                ! Turn on 2T @ mile1
&// dezone:        det 2                  ! Dezone: locale factor @ mile1
&//
&// milestone:     time .gt. 2            ! Trigger restart milestone @ mile2
&// change:        par = 2.2              ! Insert: line @ mile2
&// dezone:        det 4                  ! Dezone: locale factor @ mile2
&// dezone:        hex 2                  ! Dezone: locale factor @ mile2
```

- iRage has several pre-defined model options available.
  - The MODEL specifies the physics approximations that will be used in the xRage input deck. The physics models are indicated by *// cards.
  - The MODEL also specifies the actions that xRage will take at each of the milestones. These are indicated by &// cards.
- The iRage user also has the option to customize a MODEL named “mod” that can be called from local file space.
- The cited shortcut submits an OSO file directly to production with no further user intervention.

The MENU file may be a handy shortcut for the iRage user.

- iRage automatically generates a ".menu" file which combines the physics options specified by the MODEL with the ".mini" file produced by the the "Map" GUI.
- It may be convenient for a user to change physics options, parameters, zoning, etc. in the MENU file; and then run iRage again. For example,

  iRage menu=myjob run

  uses the information in the "myjob.menu" file to produce a new "rage.input" deck without having to call any of the GUIs. The "run" flag indicates that the job is submitted to the production queue under the assumption that the EOS and Opacity files already exist.
- Note that in the example the geometry file was not defined. iRage will build the input file; but xRage would crash.

```
Zsh ~/JAS> cat tmp1.menu
# ********************************************************************************
# ******************                tmp1.menu               ******************
# ********************************************************************************
#
*// project:      tmp1
*// pname:        210322x
*// xrage:        210104
*// cpuhrs:       36-8
*//
*// propath:      /Users/jasm-s/scratch/jasm-s/tmp1
*// geofile:      /Users/jasm-s/scratch/jasm-s/tmp1/UNKNOWN
*// eosfile:      /Users/jasm-s/scratch/jasm-s/tmp1/tmp1.eos
*// opcfile:      /Users/jasm-s/scratch/jasm-s/tmp1/tmp1.opc20
*//
*// model:        jas                    ! A model for testing purposes
*// quality:      good                   ! Default physics options and zoning
*// geometry:     2D Cylindrical         ! Dimensions / Symmetry
*// hydrobc:      Mirror                 ! Hydro Boundary: Mirror, Vacuum
*// radbc:        Vacuum                 ! Rad Boundary: Mirror, Vacuum, T(time)
*// units:        micron                 ! Mesh zoning units
*//
*// hydro:        ip6                    ! Options: ip0, ip1, ip2, ip4, ip6, vof
*// strength:     sg                     ! Options: off, ep, jc, kos, ptw, ptw1, sg
*// explosives:   cg_mie.jwl             ! Options: off, (cg,ff,hi,so,sp).(hom,jwl,mie
*// conduction:   spitzer                ! Options: off, sesame, spitzer
*// radiation:    gd                     ! Options: off, gd, mgd
*// opacity:      opc20_lte              ! Options: off, opc(20,25,30,58)_(lte,nlte)
*// plasma:       3t                     ! Options: off, 2t, 3t
*// lasers:       off                    ! Options: off, lle, nif
*// mix:          off                    ! Options: off, bhr2, bhr3, modal
*//
*// tstop:        1.0e-4                 ! Forced stop time, tmax = 1.0e-4.
*//
*// level1:       0.8192                 ! Level 1 mesh size (cm)
*// hydromesh:    0.0  130.0 -130.0  130.0    0.0    0.0    0.0 ! xmin xmax ymin ymax
*// detpoint:     0.0  -37.0    0.0  0.050    0.0000e+00        ! xdet ydet zdet Rdet
*// detpoint:    22.0  -37.0    0.0  0.050    0.0000e+00        ! xdet ydet zdet Rdet
*//
*// xensight:      grd mat rho tev prs vel
*// xparaview:     grd mat rho tev prs vel
*// tracers:      rho tev tev prs vel
```

iRage produces well-documented xRage input decks.

```
:! ** Hydro Method Options
dohydro = .true.                        ! Turn on hydrodynamics.
hydro_version = 2                       ! Current Rage hydro version
fvolpct = 0.01                          ! Mesh refinement on material gradient
rho_floor = 1.0e-6                      ! Minimum density in a cell
numrho = 6                              ! Gittings-van Leer limiter
interface_option = 1                    ! 0=No IP, 1=IP, 3=VOF
shock_detector = 2                      ! 1=IP off, 2=IP on


! ****************************************************************************
! ******************              Materials              ********************
! ****************************************************************************

numreg = 16                             ! Number of regions
nummat = 14                             ! Number of materials
ramp_num = 5                            ! Number of ramped materials
ramp_reverse(1) = 5 * .false.           ! All ramps are irreversible

cg_number = 2                           ! Number of Cerro Grande materials
cg_size = 0.0128                        ! Cerro Grande cell size
cg_dtpct = 0.4                          ! Cerro Grande time step change
cg_p_time = 0.13e-6                     ! Cerro Grand p-time change

! ****************************************************************************

! ** Material 1: BAKair
matident(1) = 'BAKair'                  ! Nickname; Rho=0.0013, Abar=14.8028
matdef( 1, 1) = 5030                    ! EOS; Rhoeos=0.0013, Abareos=14.5481
matdef(25, 1) = 0.982794                ! Scaling Ratio, SR = Abareos/Abar
matdef(61, 1) = 65030                   ! Ipcress opacity table.

! ** Map Region 1 (Background) <=> Material 1 (BAKair)
mnamereg(1) = 'Background'              ! Background air
matreg(1) = 1                           ! Region 1 <=> Material 1
rhoreg(1) = 0.0013                      ! Initial density (g/cc)
tevreg(1) = 0.025                       ! STP temperature (eV)

! Material 1 (BAKair) Initial Zoning
sizebnd(1) = 1.6384                     ! Level 1 zoning for material boundary
sizemat(1) = 1.6384                     ! Level 1 zoning for material interior
```

- iRage produces “rage.input” decks which have comments on every line.
- Well-documented input decks can serve as text books on the numerical and physical approximations available to a code.

```
! ****************************************************************************
! ******************                Start                 ********************
! ****************************************************************************

! ********** Radiation Physics **********
dorad = .true.                          ! Turn on radiation physics.

! ** Radiation Boundary Conditions: 2D Cylindrical Symmetry
tevcl = 0.0                             ! Reflective boundary left (xmin)
tevcr = 0.01                            ! Vacuum boundary right (xmax)
tevcb = 0.01                            ! Vacuum boundary below (ymin)
tevca = 0.01                            ! Vacuum boundary above (ymax)

! ** Planck vs Rosseland Mean Opacities
ipcress_do_planck_ff = .false.          ! Max(Rosseland absorption, Planck FF)
ipcress_use_pgray_mat(1) = 17 * .true.  ! T - Planck FF; F - Rosseland
do_rad_noneq_correct = .false.          ! Forego accuracy for stability
kapmin = 1.0e-20                        ! Minimum absorption coefficient (def 1e-6)

! ** Diffusion Approximation
fluxlim = .true.                        ! Levermore flux-limited diffusion
maxph4loop = 1                          ! Iterations to converge matter-rad coupling
solver_option = 50                      ! Default solver 1D/2D

! ** Insert at milestone start
par = 0.0                               ! Insert: line @ start
```

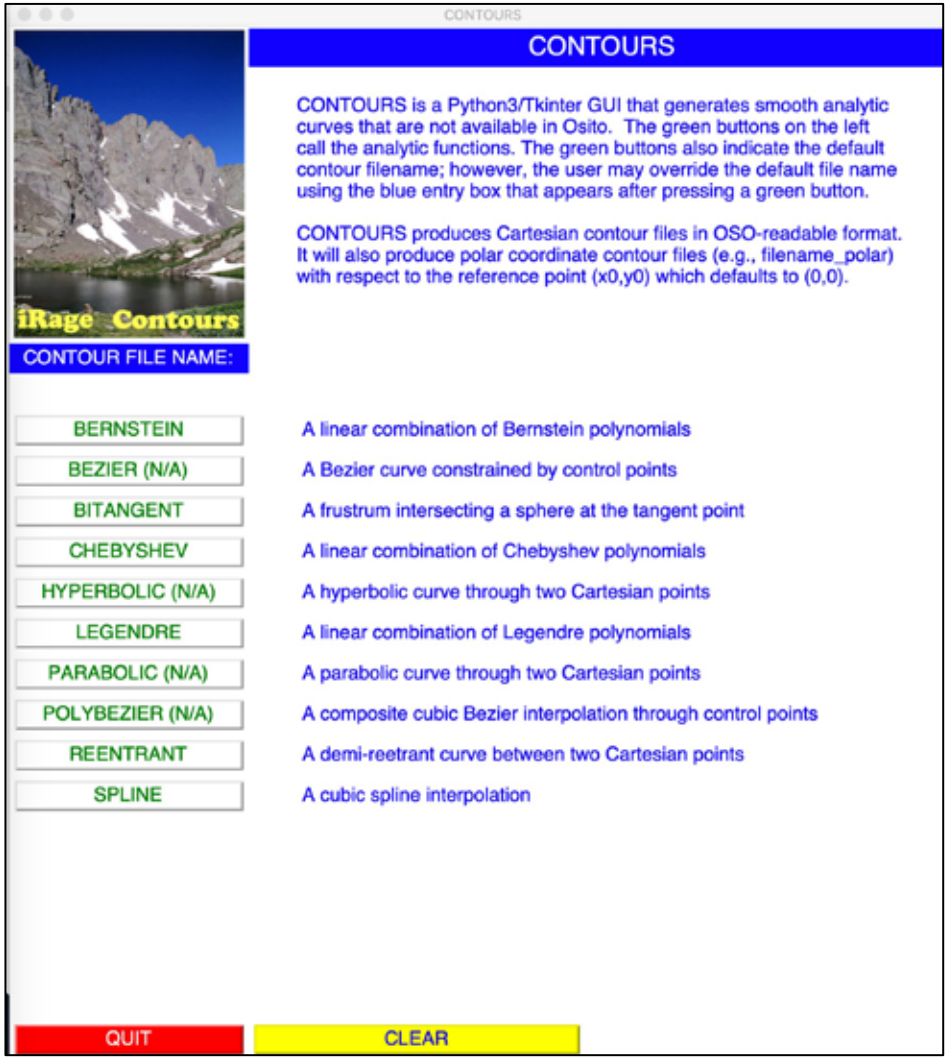
The iRage command line has a “runflag” that simplifies the submission of jobs to the production queue.

```
# ********************************************************************************
# **********************  Command Line Argument 2  ***********************
# ********************************************************************************

The iRage command line has the format:

    iRage jobflag=myjob runflag

The runflag specifies how a job is to be submitted to the production queue.

    runflag = neo    Builds the EOS tables using xRage.
                     Builds the Opacity tables using TOPS.

    runflag = new    Builds the EOS tables using xRage.
                     Builds the Opacity tables using TOPS.
                     Prepares the job for the production queue.
                     Runs the job for 10 cycles.

    runflag = now    Builds the EOS tables using xRage.
                     Builds the Opacity tables using TOPS.
                     Prepares the job for the production queue.
                     Runs the job in the production queue.

    runflag = job    Assumes the existence of EOS and Opacity tables.
                     Prepares the job for the production queue.
                     Does not submit the job to the production queue.

    runflag = run    Assumes the existence of EOS and Opacity tables.
                     Prepares the job for the production queue.
                     Submits the job to the production queue.

    runflag = walk   Assumes the existence of EOS and Opacity tables.
                     Prepares the job for the standby queue.
                     Submits the job to the standby queue.
```

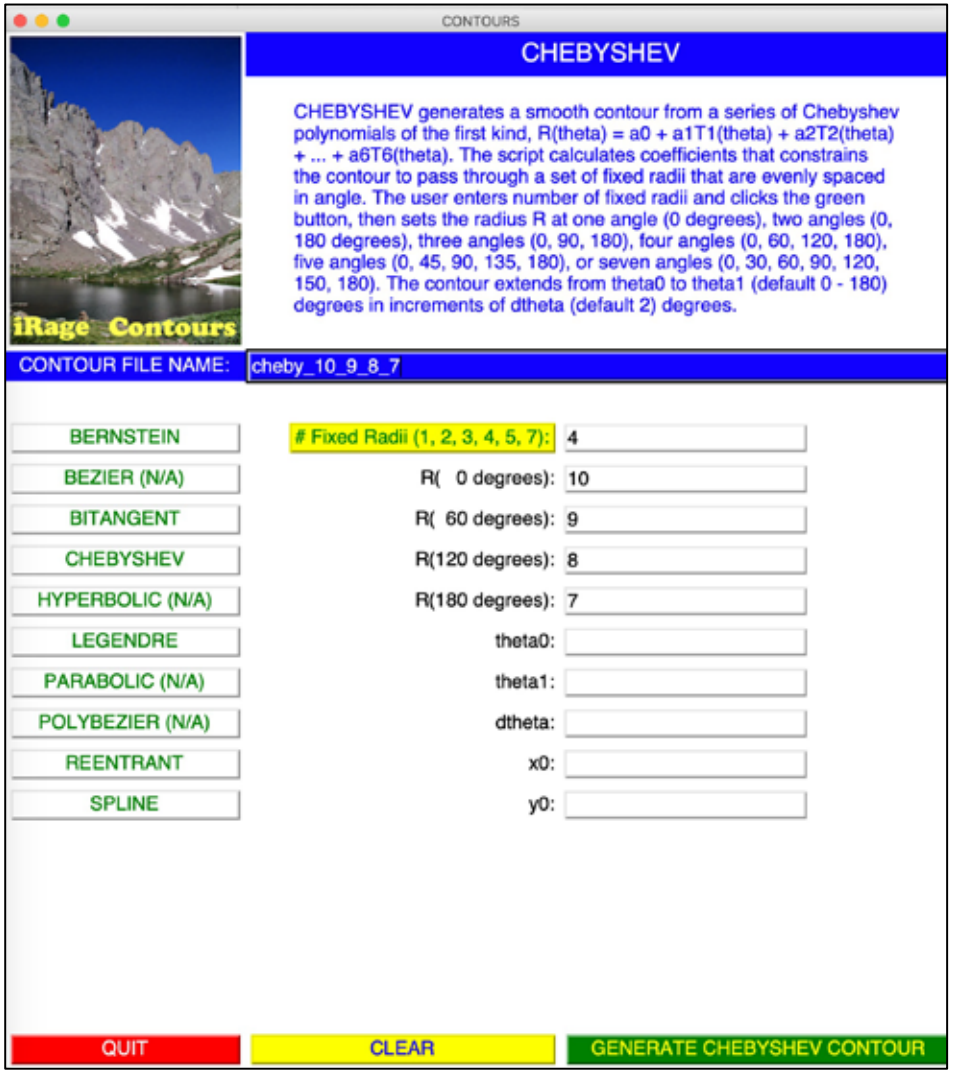
- Submitting jobs to the production queue can be a mysterious process for the novice xRage user.
  - First the user runs xRage to create EOS tables. Then the user runs TOPS to produce Opacity tables.
  - Then the user builds a fine named “run_job.csh” in order to take advantage of the EAP production scripts.
  - Then the user moves the xRage input file, the geometry file, the EOS file, the Opacity file to scratch space, and finally submits the job to the production queue.
- iRage takes care of all of this with a single flag that provides the user several options for submitting jobs.

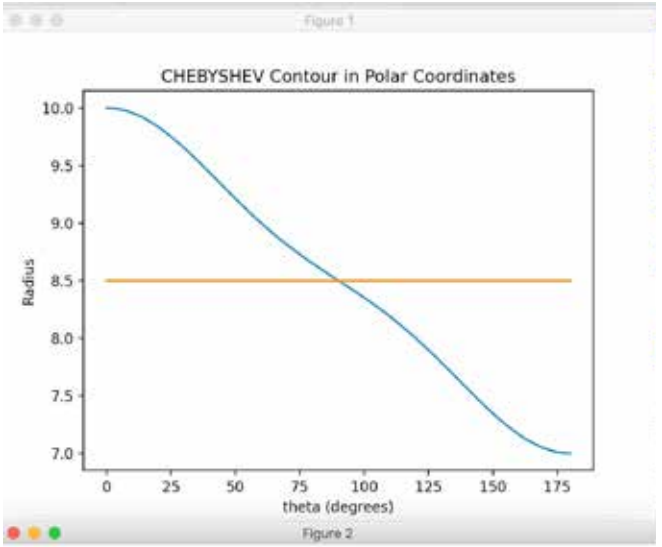
Other iRage tools: Contours (“iRage con”)

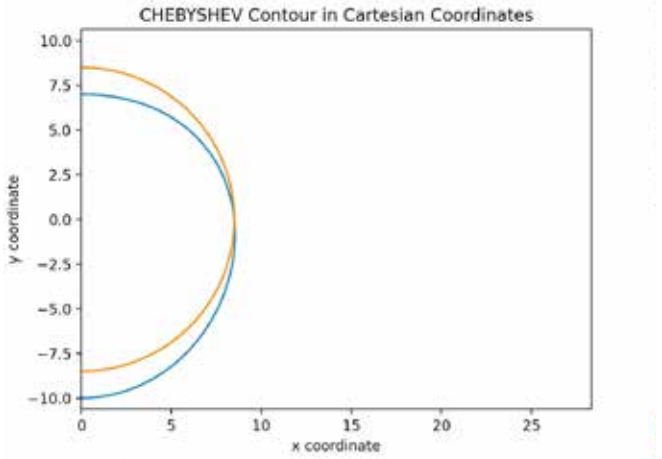
- iRage Contours provides several analytic contours that supplement the analytic functions that are available in Osito. The command “iRage contours” (or “iRage con”) calls a selection of GUIs which generate: Bernstein, Bezier, bitangent, Chebyshev, hyperbolic, Legendre, parabolic, polybezier, reentrant, and spline contours.
- The GUIs generate contours in both Cartesian and polar coordinates. The Cartesian coordinates are written in the OSO format that can be read directly by Osito and xRage.



Example of the CHEBYSHEV contour option.







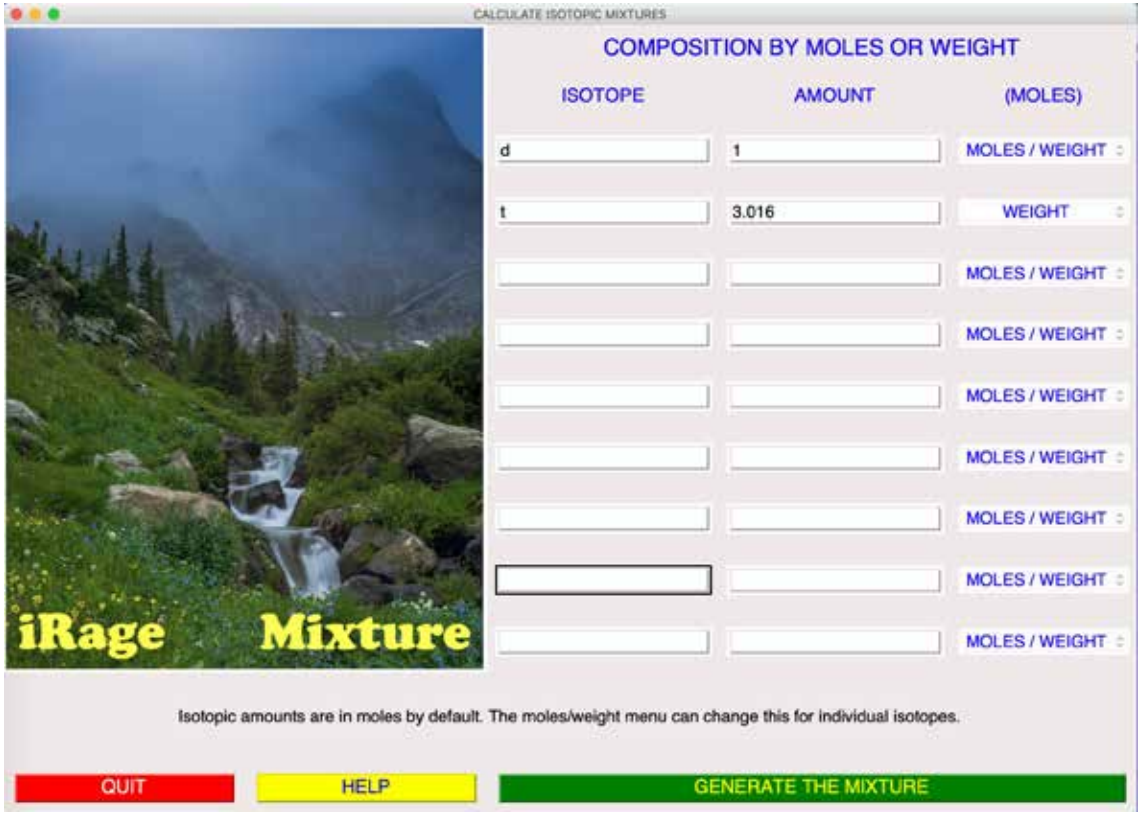
Other iRage tools: Material Lookups (e.g., “iRage steel”)

```
Zsh ~/JAS> iRage sst
iRage version 210202 @ 16:46:56

Material Name  : sst
  Sesame EOS  : 4272                    Sesame OPC  : 00000
  Sesame Abar : 55.3650                 Ipcress OPC : 64272
  Sesame Zbar : 25.8020                 Conduction  : 24279
  Sesame Rho0 : 7.8960

   Normalized Composition               Abar :  55.3621
Element   % Moles   % Weight            Zbar :  25.8030
 cr      0.202300   0.190000
 fe      0.693900   0.700000
 ni      0.103800   0.110000

Elastic-Plastic Strength Model: sst
  matdef   5 : -3.0e10                  Fail pressure (erg/cc)
  matdef   7 : 7.70e11                  Shear modulus
  matdef  12 : 3.40e9                   Yield stress
  matdef  13 : 1.0e10                   Elastic pressure

Steinberg-Guinan Flow-Stress Model: sst
  matdef 101 : 7.90                     SG reference density (g/cc)
  matdef 102 : 7.70e11                  G0, shear modulus (erg/cc)
  matdef 103 : 3.4e9                    Y0, yield strength (erg/cc)
  matdef 104 : 2.26e-12                 A, P-dep shear modulus (cc/erg)
  matdef 105 : 5.280                    B, T-dep shear modulus (1/eV)
  matdef 106 : 43.0                     beta work hardening parameter
  matdef 107 : 0.35                     n work hardening parameter
  matdef 108 : 0.00                     ep0, initial plastic strain
  matdef 109 : 25.0e9                   Ymax, Max yield strength (erg/cc)
  matdef 110 : 1.93                     Initial Gruenisen gamma
  matdef 111 : 1.4                      V-dep gamma coefficient
  matdef 112 : 0.205                    Tmelt, melt temp (eV)

iRage Finis @ 16:46:56
```

```
Zsh ~/JAS> iRage 4272
iRage version 210202 @ 16:47:30

Material Name  : sst
  Sesame EOS  : 4272                    Sesame OPC  : 00000
  Sesame Abar : 55.3650                 Ipcress OPC : 64272
  Sesame Zbar : 25.8020                 Conduction  : 24279
  Sesame Rho0 : 7.8960

   Normalized Composition               Abar :  55.3621
Element   % Moles   % Weight            Zbar :  25.8030
 cr      0.202300   0.190000
 fe      0.693900   0.700000
 ni      0.103800   0.110000

Elastic-Plastic Strength Model: 4272
  matdef   5 : -3.0e10                  Fail pressure (erg/cc)
  matdef   7 : 7.70e11                  Shear modulus
  matdef  12 : 3.40e9                   Yield stress
  matdef  13 : 1.0e10                   Elastic pressure

Steinberg-Guinan Flow-Stress Model: 4272
  matdef 101 : 7.90                     SG reference density (g/cc)
  matdef 102 : 7.70e11                  G0, shear modulus (erg/cc)
  matdef 103 : 3.4e9                    Y0, yield strength (erg/cc)
  matdef 104 : 2.26e-12                 A, P-dep shear modulus (cc/erg)
  matdef 105 : 5.280                    B, T-dep shear modulus (1/eV)
  matdef 106 : 43.0                     beta work hardening parameter
  matdef 107 : 0.35                     n work hardening parameter
  matdef 108 : 0.00                     ep0, initial plastic strain
  matdef 109 : 25.0e9                   Ymax, Max yield strength (erg/cc)
  matdef 110 : 1.93                     Initial Gruenisen gamma
  matdef 111 : 1.4                      V-dep gamma coefficient
  matdef 112 : 0.205                    Tmelt, melt temp (eV)

iRage Finis @ 16:47:30
```

- iRage performs material lookups and reverse (EOS) lookups; e.g., “iRage sst” or “iRage 4272”

Other iRage tools: Mixtures ("iRage mix")




```
iRage Finis @ 09:49:20
Zsh ~/JAS> iRage mix
iRage version 210309 @ 09:50:01
    Call the Mix GUI @ 09:50:01.

# ********************************************
# ***************  MIXTURE  ******************
# ********************************************

GUI INPUT:
 d  1  MOLES
 t  3.016  WEIGHT

Mixture Abar        2.515071
Mixture Zbar        1.000000

   Normalized Composition
Element    % Moles    % Weight
 d        0.500000    0.400400
 t        0.500000    0.599600

# ********************************************
# ********************************************

  2.5151    1.0000 %m d  0.5000 t  0.5000 %
iRage Finis @ 09:50:25
```

- "iRage mix" calls a GUI which calculates Abar, Zbar, and normalized isotopic mixtures by molar number and weight. Note that the GUI accepts input in either moles or weight for each of the isotopes in the mixture with the default being moles.

“iRage help” triggers the inline help package.

```
Zsh ~/iRAGE> iRage help
iRage version 210315 @ 19:37:32

# ********************************************************************************
# ************************  iRage Help Package  ************************************
# ********************************************************************************

   iRage is a toolkit that simplifies the process of building xRage input decks
and submitting xRage jobs to production queues.  iRage is a Zshell script that
calls several Python3/Tkinter GUIs.  In addition to building input decks, iRage
has several additional tools, including: CON, a GUI that generates several
analytic contours; MIX, a GUI that calculates Abar, Zbar, and normalizations for
isotopic mixtures; CRIB, a lookup tool that returns materials properties for a
specified material or EOS number.

   The "iRage" command is aliased by the EAP .cshrc login file, so it should be
available to Rage users; otherwise, the user can set the alias:

        alias iRage 'zsh /usr/projects/eap/tools/iRAGE/iRage.zsh'.


Help Options:
   0 - View the iRage cookbook pdf.
   1 - Command line argument 1 (jobflag)
   2 - Command line argument 2 (runflag)
   3 - GUI: Physics menu options
   4 - GUI: Detonator points
   5 - GUI: Hydro mesh
   6 - GUI: Region <=> Material mapping
   7 - GUI: Region <=> Material mapping 1D
   8 - GUI: Contours
   9 - GUI: Mixture
   a - Archive of iRage changes
   b - Mini files
   c - Menu files
   q - Quit

Help Option? 1
```

```
# ********************************************************************************
# ************************  Command Line Argument 1  *****************************
# ********************************************************************************

The iRage command line triggers several tools that guide the user through the
setup process.  The iRage command line has the format:

    iRage job(.mod)=myjob(.oso) runflag

where 'myjob' is an arbitrary name assigned by the user, (.mod) is an option
to use a pre-defined model named 'mod', and (.oso) is an option to use the
mesh and mapping information in an existing OSO file (myjob.oso). If (.oso)
is present, then iRage will generate an xRage input deck directly. Otherwise,
iRage will take the user through several GUIs.

    job = new       Calls GUIs that guide the user through the setup process.
                    Builds a 'mini' file that maps regions <=> materials.
                    Builds a 'menu' file that specifies physics options.
                    Builds an eos.inp file used to create EOS tables.
                    Builds an opc.inp file used to creat opacity tables.
                    Builds a fully commented rage.input file.

    job = mini      Assumes the existence of a file named myjob.mini
                    Calls a quicker set of GUIs.
                    Builds a 'menu' file that specifies physics options.
                    Builds an eos.inp file used to create EOS tables.
                    Builds an opc.inp file used to creat opacity tables.
                    Builds a fully commented rage.input file.

    job = menu      Assumes the existence of a file named myjob.menu
                    Skips the setup GUIs.
                    Builds an eos.inp file used to create EOS tables.
                    Builds an opc.inp file used to creat opacity tables.
                    Builds a fully commented rage.input file.
```

- iRage command line job options explained.

- iRage has help packages that can be called either from the command line or from the GUIs.

Some final comments about iRage.

- iRage is a toolkit that produces well-documented xRage input decks
- iRage can automatically submit xRage simulations to the production queue.
- iRage is an example of how the setup process can be simplified through the use of GUIs.
- iRage was designed to focus the attention of novice xRage users on selecting appropriate physics approximations rather than worrying about input deck syntax.